

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-056694

(43)Date of publication of application : 27.02.2001

(51)Int.CI.	G10L 13/00
	G06F 3/00
	G06F 3/16
	G06F 13/10
	G10L 15/22

(21)Application number : 11- (71)Applicant : DENSO CORP
232657 DENSO CREATE:KK
(22)Date of filing : 19.08.1999 (72)Inventor : ASAMI KATSUSHI
SASAKI MIKIO
ITO KENZO
HIROSE SATOSHI
MUSASHI HIKARI

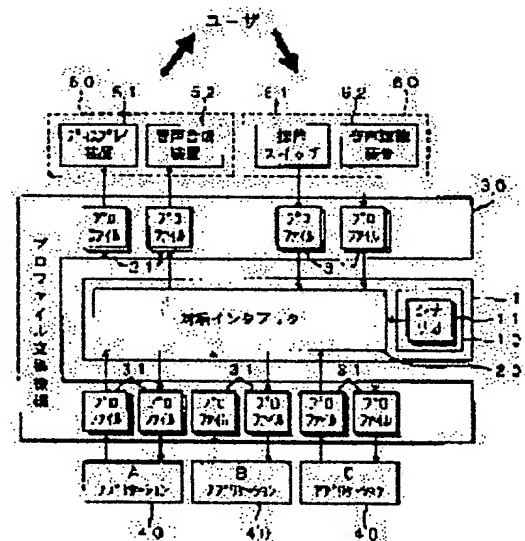
(54) INTERACTIVE USER INTERFACE DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To make it easy to design an application and to selectively execute plural applications through a series of interactions.

SOLUTION: A scenario 11 is stored in a storage device, and an interactive interpreter 20 reads this scenario out, interacts with a user according to the description of the scenario, and obtains information needed for an application.

Further, the interactive interpreter 20 decides whether or not a topic is changed, i.e., whether the input information is information related to an application other than the application being executed from the input information from the user. Then, when the topic is changed, the actuation of the application corresponding to the input information is indicated.



LEGAL STATUS

[Date of request for examination] 26.03.2003

[Date of sending the examiner's
decision of rejection]

[Kind of final disposal of application
other than the examiner's decision
of rejection or application
converted registration]

[Date of final disposal for
application]

[Patent number]

[Date of registration]

[Number of appeal against
examiner's decision of rejection]

[Date of requesting appeal against
examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

【特許請求の範囲】

【請求項1】複数のアプリケーションに対して用いられ、当該アプリケーションで必要となる情報をユーザとの対話によって取得する対話型ユーザインタフェース装置であって、

記憶手段に記憶されたシナリオを読み出し、当該読み出したシナリオの記述に従い、実行中のアプリケーションで必要な情報項目の入力要求ステップと、当該入力要求に応じてユーザから入力される入力情報の取得ステップとを有する対話処理を実行し、前記実行中のアプリケーションで必要となる情報を取得して当該アプリケーションへ送出する制御手段を備え、

さらに、前記制御手段は、前記対話処理において取得された前記入力情報が前記実行中のアプリケーション以外の別のアプリケーションに関連する情報である場合には、当該別のアプリケーションの起動指示を行うことを特徴とする対話型ユーザインタフェース装置。

【請求項2】請求項1に記載の対話型ユーザインタフェース装置において、

前記シナリオは、前記複数のアプリケーションに対応させて用意されており、

前記制御手段は、前記アプリケーションの起動指示を行うと、当該アプリケーションに対応するシナリオを前記記憶手段から読み出すことを特徴とする対話型ユーザインタフェース装置。

【請求項3】請求項1又は2に記載の対話型ユーザインタフェース装置において、

前記制御手段は、前記対話処理を実行し、前記実行中のアプリケーションで必要な情報の全てを取得した後に、当該アプリケーションへ当該情報を送出することを特徴とする対話型ユーザインタフェース装置。

【請求項4】請求項1～3のいずれか記載の対話型ユーザインタフェース装置において、

前記制御手段は、前記記憶手段から複数のシナリオを読み出した場合、当該複数のシナリオの記述に従い、前記対話処理を並行して実行可能であることを特徴とする対話型ユーザインタフェース装置。

【請求項5】請求項1～4のいずれか記載の対話型ユーザインタフェース装置において、

前記シナリオには、前記実行中のアプリケーションで必要な情報項目の入力要求を行うための要求情報、当該入力要求に応じてユーザから入力されると想定される予想入力情報、及び予想入力情報に応じた分岐情報からなる対話情報が少なくとも記述されており、

前記制御手段は、前記要求情報に基づき前記入力要求ステップにて入力要求を行い、前記取得ステップにてユーザからの入力情報を取得すると、当該取得した入力情報と前記予想入力情報とを比較し、一致する予想入力情報に対応する分岐情報に従い分岐処理を実行することを特徴とする対話型ユーザインタフェース装置。

【請求項6】請求項5に記載の対話型ユーザインタフェース装置において、

前記対話情報は、所定のグループに分けて記述されていることを特徴とする対話型ユーザインタフェース装置。

【請求項7】請求項1～6のいずれか記載の対話型ユーザインタフェース装置において、

前記ユーザとの対話は、外部のデバイスであって、ユーザへの情報要求を行うための出力デバイス及びユーザからの入力情報を取得するための入力デバイスを介して行われることを前提として、

前記制御手段は、

さらに、前記出力デバイスに依存しない出力データを当該出力デバイスに対応するデータに変換する出力データ変換手段、又は、前記入力デバイスに対応するデータを当該入力デバイスに依存しないデータへ変換する入力データ変換手段の少なくともいずれか一方を備え、

前記出力データ変換手段を備えている場合、当該出力データ変換手段を介して前記出力デバイスへのデータ出力を行い、前記入力データ変換手段を備えている場合、当該入力データ変換手段を介して前記入力デバイスからのデータを取得することを特徴とする対話型ユーザインタフェース装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ユーザとの間で自然な対話を実現し、この対話によってアプリケーションプログラムで必要な情報を取得する対話側ユーザインタフェース装置に関する。

【0002】

【従来の技術】近年、カーナビゲーション装置には、対話型のユーザインタフェースを備えるものが増えてきている。このようなカーナビゲーション装置では、ユーザとの対話を通じて情報を取得し、その情報に基づく処理を行う。例えば、音声合成装置による問いかけと、音声認識装置によるユーザの発話内容の取得とを繰り返して目的地、目的地経路上の通過地点などの情報を取得し、ルート案内を行うという具合である。なお、本明細書でいう「対話」には、上述した音声による対話だけでなく、ユーザによるスイッチ操作と装置によるディスプレイへの情報表示によってなされる情報のやり取りも含まれる。

【0003】上述したような対話処理は、ルート案内などの機能を実現するためのアプリケーションプログラムに記述されていた。そのため、アプリケーションプログラムの処理が複雑になり、その設計が困難になるという問題があった。また、アプリケーションプログラムの処理手順に対話の流れが左右されるため、自然な対話、すなわちユーザ側の自由度の高い対話を実現することが困難で、ユーザが違和感を持つことが多かった。つまり、図7に示すように、アプリケーションプログラムの中に

対話処理(項目A~Cの入力を行うための処理)が入っているため、対話処理が固定的なものになってしまうのである。

【0004】なお、以下の説明では、アプリケーションプログラムを「アプリケーション」と略して記述すると共に、アプリケーションを主語とした表現を適宜用いる。これは、実際にはCPUが主体となってアプリケーションプログラムを実行することを意味する。

【0005】上述したような問題を解決するために様々な技術が従来より提案されている。例えば特開平7-239694号公報に記載の発明は、対話の履歴に応じて自然性が損なわれない対話の範囲を判定し、常に適切な応答を行おうとするものである。また、特開平10-124087号公報には、自然な対話となるような音声入力を行い、それら音声信号の順序を入れ替えて音声認識する技術が開示されている。さらに、特開平10-111698号公報には、音声対話システムの動作の中でブロック化できるものを動作ブロックとし、動作ブロックの動作順序をシーケンスデータで記述することによって、アプリケーションの設計を容易にする技術が開示されている。

【0006】

【発明が解決しようとする課題】ところで、例えばカーナビゲーション装置が設置される車室内を例に挙げて考えると、オーディオ機器、テレビジョン受像機、自動車電話機など複数の機器が搭載される現状がある。したがって、これらの機器を例えば音声対話によって統括的に制御することが望まれている。

【0007】ところが、上述した公報に開示された技術には、あるアプリケーションに対応する対話処理という思想があり、対話の範囲は一のアプリケーションに限定されていた。このため、次のような問題が生じる。例えば、ルート案内のアプリケーションが目的地の情報を取得するための対話処理において、ユーザは自動車電話機の使用を要求することができないという問題である。つまり、上述した従来の技術では、あるアプリケーションに対する対話処理と別のアプリケーションに対する対話処理とは独立したものとなっていた。その結果、あるアプリケーションの実行中に別のアプリケーションを実行する場合、一旦、アプリケーションを終了させて、別のアプリケーションを実行させるという作業(音声によるコマンドであることも考えられる。)が必要となっており、一連の対話の中で複数のアプリケーションを選択的に実行させることができなかった。

【0008】そこで本発明は、アプリケーションの設計が容易であり、しかも、複数のアプリケーションの選択的な実行を一連の対話を通じて実現可能な対話型ユーザインタフェース装置を提供することを目的とする。

【0009】

【課題を解決するための手段及び発明の効果】上述した

目的を達成するためになされた本発明の対話型ユーザインタフェース装置は、複数のアプリケーションに対して用いられる。そして、これらアプリケーションで必要となる情報を、ユーザとの対話によって取得する。

【0010】本発明では特に、制御手段が、記憶手段に記憶されたシナリオを読み出し、当該読み出したシナリオの記述に従い、対話処理を実行し、実行中のアプリケーションで必要となる情報を取得して当該アプリケーションへ送出する。シナリオは、コンピュータファイルとして実現することが考えられる。

【0011】上述した対話処理は、実行中のアプリケーションで必要な情報の入力を要求するステップと、当該入力要求に応じてユーザから入力される入力情報を取得するステップとを有する。すなわち、この2つのステップの繰り返しによってユーザとの対話が実現される。そして、実行中のアプリケーションで必要な情報が取得される。

【0012】このように本発明では、対話処理がシナリオに記述され、アプリケーションの処理プログラムを設計する上でユーザとの対話を考慮することがないため、アプリケーションの設計を容易にすることができる。さらに、上述した制御手段は、対話処理において取得された入力情報が実行中のアプリケーション以外の別のアプリケーションに関連する情報である場合には、当該別のアプリケーションの起動指示を行う。この起動指示によってアプリケーションが起動するようにすれば、実行中のアプリケーションに必要な情報を取得するための対話処理の途中で別のアプリケーションが起動されることになる。結果として、一連の対話によって複数のアプリケーションを選択的に実行させることができる。

【0013】なお、上述したシナリオは、複数のアプリケーションに対応させて用意することが考えられる。このとき、制御手段は、アプリケーションの起動指示を行うと、当該アプリケーションに対応するシナリオを記憶手段から読み出す(請求項2)。このようにすれば、アプリケーションを新たに追加するとき、当該アプリケーションに対応するシナリオを用意すればよく、他のシナリオを変更する必要がないため、アプリケーションの追加が容易になる点で有利である。

【0014】ところで、アプリケーションで必要となる情報を自然な対話によって取得するという観点からは、請求項3に示すように、制御手段が、対話処理を実行し、実行中のアプリケーションで必要な情報の全てを取得した後に、アプリケーションへ当該情報を送出するようにすることが望ましい。

【0015】このとき、アプリケーションで必要となる情報項目が何であるかは、シナリオに記述することが考えられる。また、アプリケーションから通知されることが考えられる。アプリケーションで必要となる情報項目が固定的である場合には、シナリオに記述することが望

ましい。アプリケーションからの通知処理が不要になるからである。

【0016】このようにアプリケーションで必要となる情報を一括して対話型インタフェース装置にて取得するようにすれば、アプリケーションの処理手順に對話の流れが全く影響を受けないため、情報の取得がより自然な對話によって実現できる。なお、アプリケーション毎にシナリオを用意する場合、あるアプリケーションの実行中に別のアプリケーションが起動されると、2つのシナリオが読み出されることになる。このとき、制御手段は、新たに起動されたアプリケーションに対応するシナリオに基づく對話処理を優先して実行し、その新たに起動されたアプリケーションに必要な情報の全てを取得した後、先に実行されていたアプリケーションに対応するシナリオに基づく對話処理を再び実行することが考えられる。

【0017】しかし、ユーザの話題の転換に応じた柔軟な對話処理を実現することを考えれば、請求項4に示すように、制御手段が、記憶手段から複数のシナリオを読み出した場合、当該複数のシナリオの記述に従い、對話処理を並行して実行できるようにすることが望ましい。複数の對話処理が並行して実行できれば、ユーザからの入力情報に対し、より適切な応答ができ、より自然な對話が実現できる。

【0018】ところで、上述した對話処理を実現するためには、例えば請求項5に示すような對話情報を少なくとも記述すればよい。この對話情報は、要求情報、予想入力情報、分岐情報からなる。要求情報とは、実行中のアプリケーションで必要な情報項目の入力要求を行うための情報である。例えば音声による對話であれば、装置側の発話情報であることが考えられる。発話情報とは、例えばテキスト形式の発話文であることが考えられる。また、発話文そのものでなく、所定領域に記憶された発話文を特定する情報であってもよい。予想入力情報とは、入力要求に応じてユーザから入力されると想定される情報である。また、分岐情報とは、予想入力情報に応じた各動作への分岐を示す情報である。

【0019】このとき制御手段は、入力要求ステップにて要求情報に基づく入力要求を行い、取得ステップにてユーザからの入力情報を取得すると、その取得した入力情報と予想入力情報とを比較し、一致する予想入力情報に対応する分岐情報に基づく分岐処理を実行する。分岐情報には、別の對話情報への分岐を指示する情報が含まれる。その場合、制御手段は、分岐先の對話情報の要求情報に基づきユーザへの入力要求を行う。また、分岐先の各動作には、上述したようなアプリケーションの起動指示が考えられる。

【0020】このようにユーザから入力されると想定される情報を予想入力情報として記述し、それら予想入力情報に対する動作情報を記述すれば、ユーザ入力に対す

る装置側の応答に柔軟性を持たせることができ、より一層自然な對話が実現できる。また、シナリオの記述が簡単になり、シナリオの設計も容易になる。

【0021】なお、シナリオの設計を容易にするという観点からは、請求項6に示すように、所定のグループに分けて對話情報を記述するとよい。例えば目的地の入力に関するグループ、経由地の入力に関するグループという具合にグループ分けして記述する。このようにすれば、對話情報から對話情報への分岐を設計するにあたり、その分岐先を設計者が把握しやすくなり、結果として、シナリオの設計が容易になる。なお、グループには、階層構造を持たせるようにしてもよい。例えば、上述した目的地の入力に関するグループと経由地の入力に関するグループとをまとめてルート案内に関する情報を入力するためのグループとして記述するという具合である。

【0022】ところで、上述したように「對話」は、音声による對話には限られない。したがって、音声だけでなく、操作スイッチによってもユーザからの情報が入力されることが考えられる。このような場合、入力デバイスが複数存在することになり、各入力デバイスに対応するプログラミングを行う必要が生じてくる。この問題は、出力デバイスが複数存在する場合も同様に生じる問題である。

【0023】このように外部のデバイス、すなわちユーザへの情報要求を行うための出力デバイス及びユーザからの入力情報を取得するための入力デバイスを介してユーザとの對話を行う前提に立てば、請求項7に示すように、制御手段がさらに、出力データ変換手段又は入力データ変換手段の少なくとも一方を備える構成とすることが考えられる。

【0024】出力データ変換手段は、出力デバイスに依存しない出力データを出力デバイスに対応するデータへ変換する。一方、入力データ変換手段は、入力デバイスに対応するデータを入力デバイスに依存しないデータへ変換する。出力データ変換手段を備える構成であれば、制御手段は、この出力データ変換手段を介して出力デバイスへデータを出力する。同様に、入力データ変換手段を備える構成であれば、制御手段は、この入力データ変換手段を介して入力デバイスからのデータを取得する。このようにすれば、出力データ変換手段及び入力データ変換手段がデバイスの違いによるデータ形式の違いを吸収するため、對話処理を実現するプログラムの中でデバイスの違いを考慮する必要がなくなり、処理プログラムの設計が容易になる。

【0025】

【発明の実施の形態】以下、本発明を具体化した一実施例を図面を参照して説明する。図1は、本発明を具体化した対話型ユーザインタフェース装置（以下「對話装置」という。）1を備えた對話システムのソフトウェア

構成を示す説明図である。実施例の対話システムは、CPU、ROM、RAM及びI/Oを有するコンピュータシステムとして構成されている。

【0026】対話システムは、対話装置1、プロファイル交換機構30、アプリケーション40、出力デバイス50、入力デバイス60とを備えている。対話装置1は、プロファイル交換機構30を介し、アプリケーション40、出力デバイス50及び入力デバイス60との間で情報交換を行う。プロファイル交換機構30には、交換する情報を記述するための複数のプロファイル31が記憶されている。

【0027】例えばアプリケーション40から対話装置1への情報の伝達は、次のようにして行われる。まずアプリケーション40が、伝達する情報を所定のプロファイル31に記述する。続いて対話装置1が、この所定のプロファイル31に記述された情報を参照する。これによって、アプリケーション40から対話型ユーザインタフェース装置1への情報の伝達がなされる。

【0028】なお、対話装置1からアプリケーション40への情報の伝達も同様に所定のプロファイル31を介してなされる。また、対話装置1と出力デバイス50、入力デバイス60との間でも同様の情報伝達が行われる。このようなプロファイル交換機構30を介した情報伝達を行うことによって、対話装置1、アプリケーション40、出力デバイス50、入力デバイス60の各処理プログラムは、プロファイル31のみに基づいて設計することができ、他の処理プログラムを考慮する必要がないため独立性が高くなり、追加及び変更などが容易になるという点で有利である。なお、この技術については、特願平11-231528号に詳しく開示した。

【0029】上述した対話装置1は、記憶装置10と、対話インタプリタ20とを備えている。記憶装置10は、例えばRAMなどで実現してもよいし、ハードディスク装置などで実現してもよい。この記憶装置10には、シナリオ11が記憶されている。対話インタプリタ20は、このシナリオ11を読み出し、このシナリオ11の記述に従い動作する。

【0030】出力デバイス50としては、ディスプレイ装置51及び音声合成装置52が接続されている。ディスプレイ装置51は、表示にてユーザへ情報入力を要求するためのものであり、音声合成装置52は、合成音声にてユーザへ情報入力を要求するためのものである。

【0031】一方、入力デバイス60としては、操作スイッチ61及び音声認識装置62が接続されている。操作スイッチ61は、ユーザがスイッチ操作にて情報を入力するためのものであり、音声認識装置62は、ユーザが音声にて情報を入力するためのものである。

【0032】また、アプリケーション40は、所定機能を実現するアプリケーションプログラムである。本実施例で図1中に示すアプリケーション40は、ソフトウェア的なまとまりを示している。また、アプリケーション40は複数存在している。以下の説明でアプリケーション40を区別する際、図1中に示すA～Cの記号を付して記述する。

【0033】このように構成された本実施例の対話システムでは、シナリオ11の記述に基づき、アプリケーション40からの要求に応じ、そのアプリケーション40で必要となる情報を、出力デバイス50及び入力デバイス60を介したユーザとの対話にて取得し、アプリケーション40へ通知する。このような動作を行うのが対話インタプリタ20である。ただし、対話インタプリタ20も上述したアプリケーション40と同様にソフトウェアであり、対話インタプリタ20が動作するというのは、実際には、コンピュータシステムとして構成される本対話システムの備えるCPUにより対話インタプリタ20として示した処理プログラムが実行されることを意味している。

【0034】そこで次に図2の説明図に基づき、対話インタプリタ20について説明する。対話インタプリタ20の処理は、上位処理である親タスクと、下位処理である子タスクとからなっている。したがって、対話インタプリタ20は、親タスクを実現するための親タスクプログラム21と、子タスクを実現するための子タスクプログラム22とから構成されている。

【0035】親タスクプログラム21は、プロファイルインタフェース21a、入力フィルタ21b及びタスク間メッセージ管理機構21cを備えている。プロファイルインタフェース21aは、上述したプロファイル交換機構30にアクセスし、プロファイル31に記述された情報を参照したり、更新したりするためのプログラムである。

【0036】入出力フィルタ21bは、対話インタプリタ20の子タスクプログラム22を、出力デバイス50及び入力デバイス60に依存させないためのデータ変換プログラムである。例えば本実施例では、入力デバイス60として操作スイッチ61及び音声認識装置62が接続されている。このとき、操作スイッチ61を介して入力されプロファイル31に記述される操作データと、音声認識装置62を介して入力されプロファイル31に記述される音声データとは、同一情報を示すものであっても、異なる形式のデータとなる。したがって、プロファイルインタフェース21aを介して入力される入力データを、入出力フィルタ21bが入力デバイス60に依存しないデータに変換することによって、子タスクプログラム22では、入力デバイス60の違いを考慮する必要がなくなる。これは、出力デバイス50についても同様である。例えばディスプレイ装置51に対する表示データと、音声合成装置52に対する発話データとは、同一の情報であっても、異なるデータ形式で書き込まれる。したがって、入出力フィルタ21bが出力デバイス50

に合わせたデータ出力を行うことによって、子タスクプログラム22では、出力デバイス50の違いを考慮する必要がなくなる。

【0037】タスク間メッセージ管理機構21cは、複数の子タスクプログラム22との間でメッセージのやり取りをするプログラムである。本実施例の対話インタプリタ20では、このような構成を採用することにより、複数の子タスクプログラム22が並行して実行できるようになっている。

【0038】一方、子タスクプログラム22は、パーザ22a及びパフォーマンス実行関数22bを備えている。子タスクプログラム22は、対話装置1の記憶装置10に記憶されたシナリオ11を読み出し、このシナリオの記述に従う処理を実行するためのものである。このときシナリオ11の記述を解釈するプログラムが、パーザ22aである。パーザ22aによって解釈された内容に基づき所定の処理を実行するためのプログラムが、パフォーマンス実行関数22bである。

【0039】次に、本対話装置1の記憶装置10に記憶されるシナリオ11について説明する。シナリオ11は、上述したアプリケーション40に対応させて用意される。したがって、アプリケーション40の起動時に、そのアプリケーション40に対応するシナリオ11が読み出される。なお、アプリケーション40が一つも実行されていないときは、基底シナリオと呼ばれるシナリオ11が読み出される。

【0040】図3に示すように、記憶装置10に記憶されたシナリオ11は、シナリオ名で区切られる。具体的には、図3中の「scenario」が一つのシナリオ11の開始を示し、「/scenario」がそのシナリオ11の終了を示す。つまり、「scenario (シナリオ名)」から「/scenario (シナリオ名)」までが一つのシナリオ11に対応するのである。(シナリオ名)には、各シナリオ11を区別するためのユニークな名前が記述される。

【0041】シナリオ11は、パフォーマンスと呼ばれる所定動作を示す記述の集合となっている。このパフォーマンスは、「シーン」と呼ばれる複数のグループのいずれかに属する。各「シーン」グループは、「scene (シーン名)」の記述から「/scene (シーン名)」の記述で表される(図中では、一つの「シーン」グループを記号②で示した。)

【0042】そしてさらに、各「シーン」グループは、「アクト」と呼ばれる複数のグループのいずれかに属する。各「アクト」グループは、「act (アクト名)」から「/act (アクト名)」の記述で表される(図中では、一つの「アクト」グループを記号①で示した。)

【0043】このようにパフォーマンスは、複数階層にグループ化されて記述される。上述したパフォーマンスには、対話インタプリタ20の動作が記述される。例えばパフォーマンスに、所定のアプリケーション40の実

行指示が記述されている場合、対話インタプリタ20は、その記述に従い、そのアプリケーション40の実行指示を行うという具合である。また、パフォーマンスには、ユーザとの対話を実現するための対話パフォーマンスが含まれる。対話パフォーマンスは、図4に示すように、装置側の質問文と、それに対して予想されるユーザの回答と、その回答に対する他のパフォーマンスへの分岐情報が記述される。

【0044】本実施例の対話システムが、ユーザとの対話によって、アプリケーション40で必要となる情報を取得するシステムであることは上述した。したがって、装置側の質問文とは、アプリケーション40で必要な情報の入力をユーザに対して要求するものであり、これに対するユーザからの入力情報を取得するのである。この質問文は、例えばテキスト形式の文章データとしてもよいが、別の領域に記憶した文章データのいずれかを特定するデータとしてもよい。また、ユーザの入力情報が質問文に沿ったものであるとは限られない。したがって、予想されるユーザの回答は複数種類記述する。そして、ユーザからの回答に応じて適切な応答を可能にするよう、予想されるユーザの回答に対応させて各パフォーマンスへの分岐情報を記述する。

【0045】次に、上述したように構成された本対話システムの動作を説明する。ここでは、最初に対話インタプリタ20のシナリオ11に基づく動作を図2を用いて説明し、その後、システム全体の動作を図5に示したフローチャートを参照して説明する。

【0046】本実施例の対話システムが起動されると、対話インタプリタ20が起動され、親タスクプログラム21が実行されて親タスクが開始される。親タスクでは、上述したタスク間メッセージ管理機構21cを介しメッセージが送出され、子タスク(第1タスク)プログラム22の実行指示がなされる。すると、子タスクが開始され、子タスクにてシナリオ11が読み出される。ここで読み出されるシナリオ11が上述した基底シナリオである。

【0047】子タスクでは、上述したシナリオ11(基底シナリオ)のパフォーマンスをパーザ22aによる処理にて解釈し、パフォーマンス実行関数22bによってそのパフォーマンスが実行される。そして、その実行結果がタスク間メッセージ管理機構21cに対して出力される。親タスクでは、実行結果に基づく処理が行われる。なお、子タスクでは、基本的にシナリオ11に記述された順序でパフォーマンスが順次実行されていく。ここでパフォーマンスに分岐情報があると、その分岐情報に基づき、分岐先のパフォーマンス以降のパフォーマンスが実行されることになる。

【0048】ここで子タスクで対話パフォーマンスに基づく処理が実行される場合を例に挙げて、さらに詳細に説明する。対話パフォーマンスである場合、子タスクで

は、装置側の質問文が読み出され、親タスクへ通知される。親タスクでは、入出力フィルタ21bによってデータ変換を行い、プロファイルインタフェース21aによって、音声合成に関するプロファイル31を更新する。すると、出力デバイス50の音声合成装置52が、この音声合成に関するプロファイル31を参照して、指示された装置側の質問文に基づく発話を行う。なお、表示に関するプロファイル31に質問文を書き込むようにし、ディスプレイ装置51に質問文が表示されるようにしてもよい。

【0049】この装置側の問いかけに対するユーザの発話があると、入力デバイス60の音声認識装置62がその発話に対する認識結果を音声認識に関するプロファイル31へ書き込む。また、装置側の問いかけに対するユーザのスイッチ操作があると、入力デバイス60の操作スイッチ61がその操作情報を操作に関するプロファイル31へ書き込む。

【0050】対話インタプリタ20の親タスクでは、音声認識に関する又は操作に関するプロファイル31の更新を判断し、プロファイルインタフェース21aによる処理にてプロファイル31に記述されたデータを読み出す。このデータは、入力デバイス60に依存したデータであるため、入出力フィルタ21bによる処理で入力デバイス60に依存しない形式のデータに変換される。そして、この変換された入力情報（データ）は、タスク間メッセージ管理機構21cを介し、実行中の子タスクへ通知される。

【0051】子タスクでは、通知された入力情報と予想されるユーザ回答とを比較し、対応する分岐情報を読み出し、分岐情報に示されるパフォーマンスへの分岐を行う。ここで分岐先のパフォーマンスが対話パフォーマンスであれば、その対話パフォーマンス中の装置側の質問文を読み出し、親タスクへ通知する。これによって、上述したユーザとの対話が繰り返される。このような対話パフォーマンスから対話パフォーマンスへの分岐を前提として、上述したパフォーマンスのグループ分けは意味的な階層構造となっている。例えばルート案内のアプリケーション40に対応するシナリオ11の対話パフォーマンスには、目的地を入力させるための対話動作や経路地を入力させるための対話動作が記述される。そこで例えば、目的地入力のための対話パフォーマンスの集合、経路地入力のための対話パフォーマンスの集合がそれぞれ、上述した「シーン」グループを構成するようにする。そして、これら2つの「シーン」グループが、上述した「アクト」グループを構成するようにする。このようにすれば、設計者が分岐先を容易に把握できることになる。一方、分岐先のパフォーマンスが対話パフォーマンスでなければ、そのパフォーマンスを実行し、その実行結果を親タスクへ通知する。

【0052】続いて図5のフローチャートに基づき、対

話インタプリタ20の処理を具体的に説明する。なお、図5中に第1タスク、第2タスク及び第3タスクとして示したフローチャートが対話インタプリタ20にて実行される処理である。なお、S100、S110、S120、S160、S170の処理は第1タスクにて実行され、S130、S140、S150、S200、S210、S220の処理は第2タスクにて実行され、S180、S190の処理は第3タスクにて実行される。第1～3のタスクは上述した子タスクに相当する。また、プロファイル31の参照/更新処理は、親タスクにて実行される処理に相当する。

【0053】まず最初のステップS100において、基底シナリオがロードされる。この処理は、対話インタプリタ20が記憶装置10に記憶されたシナリオ11の中の基底シナリオを読み出すものである。基底シナリオ中に記述されたパフォーマンスの実行によって、続くS110では、ユーザ要求の受け付けを行う。この処理は、プロファイル31を更新することで出力デバイス50を介してユーザの要求が何であるかを問いかけ、入力デバイス60を介して入力されるユーザの入力情報をプロファイル31を介して取得するものである。

【0054】そして、次のS120では、入力情報に応じたアプリケーション40の起動指示を行う。この処理は、プロファイル31に対して起動指示を書き込むものであり、システムによってその起動指示の対象となっているアプリケーション40が起動される。なお、ここで起動されたアプリケーション40が図1中のAアプリケーション40であるとして、以下の説明を続ける。

【0055】アプリケーション40の起動を指示すると、対話インタプリタ20の親タスクでは、新たな子タスク（第2タスク）の起動指示がなされ、この子タスクにて、Aアプリケーション40に対応するシナリオ11がロードされる（S130）。次のS140では、Aアプリケーション40の要求を受け付ける。この要求は、所定の情報項目の取得要求である。Aアプリケーション40が要求する所定の情報項目は、上述したS120の起動指示によってAアプリケーション40が起動されると、Aアプリケーション40によってプロファイル31に対して書き込まれる。

【0056】続くS150では、ユーザからの入力を受け付ける。この処理は対話パフォーマンスの実行処理であり、ここでは、対話パフォーマンス中にある装置側の質問文をプロファイル31に書き込む。その結果、出力デバイス50によってユーザに対する問いかけが行われる。また、ここでは、入力デバイス60を介したユーザの入力情報をプロファイル31を介して取得する。

【0057】ここで、ユーザからの入力情報は、実行中の2つの子タスクに通知される。基底シナリオに基づく子タスク（第1タスク）では、ユーザの入力情報に基づいて話題の転換があったか否かを判断する（S16

0)。ここでいう話題の転換とは、例えばルート案内における目的地設定のための問いかけに対し、「CDを再生したい」というような他のアプリケーションに関連する回答があった場合をいう。ここで話題の転換があったと判断された場合 (S160: YES)、S170にて、話題に対応する別のアプリケーション40の起動指示を行う。この起動指示により起動されたアプリケーション40を図1中のBアプリケーション40として以下の説明を続ける。一方、話題の転換がないと判断された場合 (S160: NO)、S200へ移行する。

【0058】アプリケーションの起動指示があると、対話インタプリタ20の親タスクでは、新たな子タスク (第3タスク) の起動指示がなされる。その結果、S180では、その子タスクにて、Bアプリケーション40に対応するシナリオ11がロードされる。そして、このシナリオ11の記述に従い対話処理が実行され、Bアプリケーション40で必要な情報が取得されると、続くS190では、シナリオ終了処理を行う。この処理は、Bアプリケーション40に対応して記憶装置10から例えばRAM (不図示) 上に読み出されたシナリオ11を、そのRAM上から削除するものである。その後、第3タスクは終了する。

【0059】Bアプリケーション40が起動された場合にはS190から移行する、又はBアプリケーション40が起動されなかった場合にはS160から移行する。S200では、Aアプリケーション40で必要な情報が全て取得されたか否かを判断する。ここで必要な情報項目のうちで未取得のものがある場合 (S200: NO)、S150からの処理を繰り返す。一方、必要な情報項目が全て取得されている場合 (S200: YES)、S210へ移行する。

【0060】S210では、情報の引き渡しを行う。この処理は取得した情報をプロファイル31に書き込む処理であり、アプリケーションは、それらの情報をプロファイル31を参照して取得する。続くS220では、シナリオ終了処理を行う。この処理は、Aアプリケーション40に対応するシナリオ11を例えばRAM上から削除するものである。その後、第2タスクを終了する。

【0061】これによって基底シナリオに基づく処理が再び実行され、特にユーザからシステム終了の指示が出ていなければ (S230: NO)、S110へ移行し、ユーザ要求受付 (待ち) 状態となる。一方、システム終了指示が出ていれば (S230: YES)、対話インタプリタ20は処理を終了する。

【0062】なお、S180の後に示した「対話処理」は、詳しくは、S140、S150、S160、S170、S200、S210と同様の処理である。ところで、上述した処理の中において、対話インタプリタ20は、アプリケーション40からの要求に応じて、そのアプリケーション40で必要となる情報を一括して取得す

る点を特徴としている。そこで次に図6の説明図に基づき、これについて説明する。

【0063】図6は、図5中のS150、S200、S210の処理ステップを詳しく示した説明図である。アプリケーション40からプロファイル31を介した所定の情報項目の通知があると、まず未入力項目に応じた問いかけを行い (S151)、続いてユーザの応答入力を受け付け (S152)、次に入力内容を要求項目に割り振る (S153)。そして、全ての項目が満たされたか否かを判断し (S200)、満たされている場合は (S200: YES)、情報の引き渡しを行う (S210)。一方、満たされていない場合は (S200: NO)、S151からの処理を繰り返す。

【0064】このように、本実施例の対話システムでは、ユーザとの対話処理には、アプリケーション40は全く関与していない。つまり、アプリケーション40の処理手順に対話処理の流れが何等影響されないのである。なお、本実施例における対話装置1の記憶装置10が「記憶手段」に相当し、対話インタプリタ20の処理を実行する対話システムのCPUが「制御手段」及び「変換手段」に相当する。

【0065】次に、本実施例の対話システムが発揮する効果を説明する。本実施例の対話システムでは、シナリオ11を記憶装置10に記憶するようにし、対話インタプリタ20は、このシナリオ11を読み出し (図5中のS130)、このシナリオ11の記述に従ってユーザとの対話を行い (S150)、アプリケーション40で必要な情報を取得する (S150)。したがって、アプリケーション40の設計にあたって、対話に関するシナリオ11がアプリケーション40とは別になっているため、アプリケーション40の設計を容易にすることができる。

【0066】しかも、対話インタプリタ20は、ユーザからの入力情報に基づき、話題の転換があったか否か、すなわち、入力情報が実行中のアプリケーション以外の別のアプリケーションに関連する情報であるか否かを判断する (S160)。そして、話題の転換があった場合には (S160: YES)、入力情報に応じた別のアプリケーションの起動指示を行う (S170)。したがって、実行中のアプリケーション40に必要な情報を取得するための対話処理の途中で別のアプリケーション40を起動できる。結果として、一連の対話によって複数のアプリケーション40を選択的に実行させることができる。

【0067】また、本実施例の対話システムでは、シナリオ11をアプリケーション40毎に対応させて用意したため、アプリケーション40を新たに追加するとき、当該アプリケーション40に対応するシナリオ11だけを追加すればよく、他のシナリオ11を変更する必要がないため、アプリケーション40の追加が容易になる点

で有利である。

【0068】そして、このシナリオ11には、対話処理を実現するための対話パフォーマンスが記述される。この対話パフォーマンスは、装置側の質問文、予想されるユーザの回答、及びその回答に基づく他のパフォーマンスへの分岐情報という簡単な記述となっている(図4参照)。したがって、シナリオ11の設計や変更も容易である。

【0069】また、対話パフォーマンスを含むシナリオ11のパフォーマンスは、意味的なグループ分けがなされて記述されている(図3参照)。例えばルート案内のアプリケーション40に対応するシナリオ11の対話パフォーマンスには、目的地を入力させるための対話動作や経由地を入力させるための対話動作が記述される。そこで例えば、目的地入力のための対話パフォーマンスの集合、経由地入力のための対話パフォーマンスの集合がそれぞれ、上述した「シーン」グループを構成するようにする。そして、これら2つの「シーン」グループが、上述した「アクト」グループを構成するようにする。このようにすれば、設計者が分岐先を容易に把握できる。つまり、上述した分岐情報を記述するにあたって、設計者は、分岐先のパフォーマンスを把握し易く、この意味においてもシナリオ11の設計/変更が容易になっている。

【0070】さらにまた、本実施例の対話システムでは、アプリケーション40で必要となる情報を対話インタプリタ20が一括して取得する(図6参照)。本実施例の対話システムでは、ユーザとの対話処理には、アプリケーション40は全く関与していない。つまり、対話処理の流れは、アプリケーション40の処理手順に何等影響されないものである。これによって、自然な対話によってアプリケーション40の情報が取得できる。

【0071】さらに、読み出された複数のシナリオ11のパフォーマンスを子タスクでは、並行して実行可能になっていることも、自然な対話処理の実現に寄与する結果となっている。一つのシナリオ11に基づく一律の対話処理に対して対話に幅を持たせることができるためである。

【0072】加えて、上述したシナリオ11の対話パフォーマンスには、予想されるユーザの回答とそれに対応する分岐情報が記述されるが、ユーザの入力情報が質問文に沿ったものであるとは限られない。したがって、予想されるユーザの回答として複数種類の回答を記述し、予想されるユーザの回答に対応させて、各パフォーマンスへの分岐情報を記述した。これによって、ユーザの入力情報に応じた適切な応答ができ、より自然な対話が実現できる。

【0073】また、本実施例の対話インタプリタ20を構成する親タスクプログラム21は、入出力フィルタ21bを備えている。そして、この入出力フィルタ21b

による処理にて、プロファイルインタフェース21aを介して入力される入力データが、入出力フィルタ21bが入力デバイス60に依存しないデータに変換されるため、子タスクプログラム22では、入力デバイス60の違いを考慮する必要がなくなる。これは、出力デバイス50についても同様である。これによって、対話インタプリタの設計も容易になっている。

【0074】以上、本発明はこのような実施例に何等限定されるものではなく、本発明の主旨を逸脱しない範囲において種々なる形態で実施し得る。上記実施例では、アプリケーション40から必要な情報項目の取得要求を受け付け(図5中のS140)、それら情報項目に対応する情報を、対話インタプリタ20にて取得するものであった(S150)。これに対して、アプリケーション40で必要な情報項目が固定的である場合には、予めそのアプリケーション40に対応するシナリオ11に、それら情報項目を記述しておくようにしてもよい。この場合、アプリケーション40から対話インタプリタ20への要求の通知処理、例えば図5中のS140の要求受付処理が必要なくなる点で有利である。

【図面の簡単な説明】

【図1】実施例の対話システムのソフトウェア構成を示す説明図である。

【図2】対話システムの対話インタプリタのソフトウェア構成を示す説明図である。

【図3】シナリオの記述を例示する説明図である。

【図4】シナリオに記述される対話パフォーマンスの内容を示す説明図である。

【図5】対話インタプリタ処理を示すフローチャートである。

【図6】対話インタプリタ処理の特徴部分を示すための説明図である。

【図7】従来の対話処理を示す説明図である。

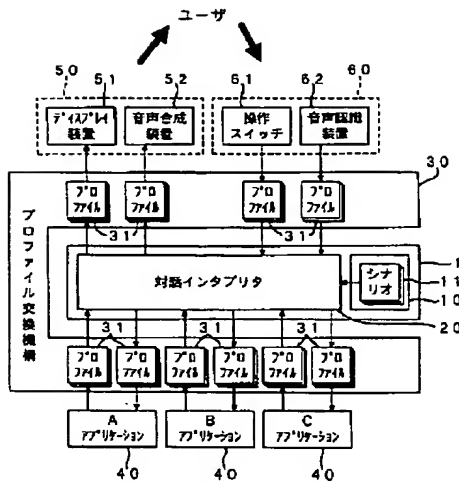
【符号の説明】

1…対話装置	10…記憶装置
11…シナリオ	20…対話インタプリタ
21…親タスクプログラム	21a…プロファイルインタフェース
21b…入出力フィルタ	21c…タスク間メッセージ管理機構
22…子タスク	22a…パーザ
22b…パフォーマンス実行関数	30…プロファイル交換機構
31…プロファイル	40…アプリケーション
50…出力デバイス	51…ディスプレイ装置
52…音声合成装置	60…入力デバイス

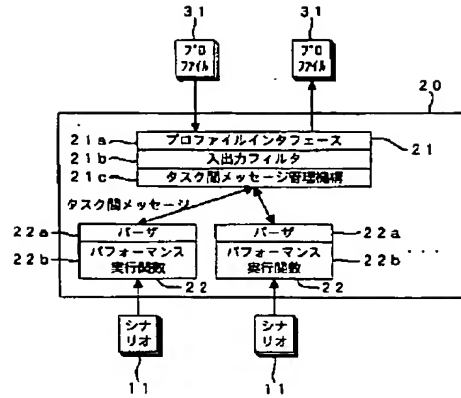
61...操作スイッチ

62...音声認識装置

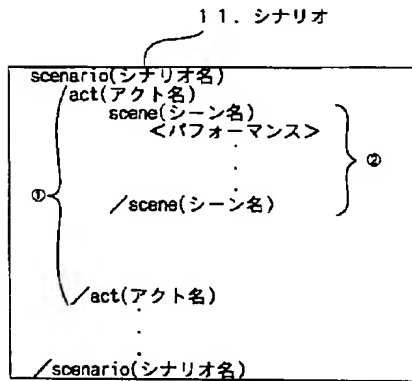
【図1】



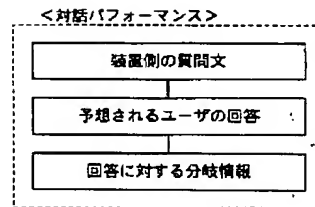
【図2】



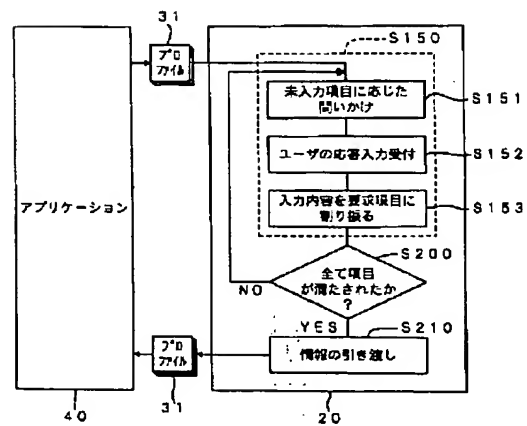
【図3】



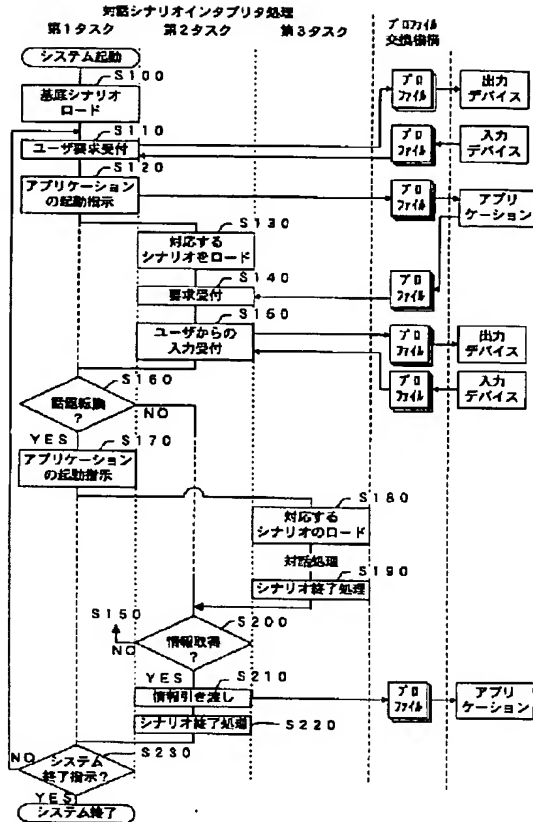
【図4】



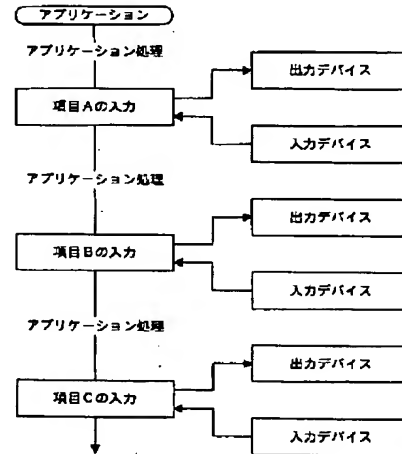
【図6】



【図5】



【図7】



フロントページの続き

(51) Int. Cl.⁷
G10L 15/22

識別記号

FI
G10L 3/00ノート (参考)
561D

- (72) 発明者 笹木 美樹男
愛知県刈谷市昭和町1丁目1番地 株式会
社デンソー内
- (72) 発明者 伊藤 健三
愛知県名古屋市中区錦二丁目15番20号 株
式会社デンソークリエイト内
- (72) 発明者 ▲廣▼瀬 智
愛知県名古屋市中区錦二丁目15番20号 株
式会社デンソークリエイト内

- (72) 発明者 八道 光
愛知県名古屋市中区錦二丁目15番20号 株
式会社デンソークリエイト内
- Fターム(参考) 5B014 EA01 EB03 FA06 FB03
5D015 LL02 LL05 LL06 LL10
5D045 AB17
5E501 AA22 AB30 AC03 AC37 BA05
EA10 EA21 FA06 FA50
9A001 DD07 DD11 HH17 HH18